REMARKS

Claims 1, 14, and 20 have been amended. No claims have been added or cancelled. Therefore, claims 1-31 are pending in the application. Reconsideration is respectfully requested in light of the following remarks.

Section 102(b) Rejections:

The Examiner rejected claims 1-3, 5, 6, 8-22, 24, 25 and 27-31 under 35 U.S.C. § 102(b) as being anticipated by Hughes (WO 01/35212). Applicants traverse this rejection for at least the following reasons.

Regarding claim 1, contrary to the Examiner's assertion, Hughes fails to teach or suggest *the load store unit is configured to generate an index dependent on at least a portion of an address of a load operation, to index into the indexed STLF buffer using the generated index to select one of the plurality of entries, and to forward data included in the one of the plurality of entries as a result of the load operation.* The Examiner cites page 34, lines 21-30 and claims 1-2 as teaching these limitations. Applicants assert that the Examiner's citations do not teach the load store unit is configured to generate an index, or to index into the indexed STLF buffer using the generated index to select one of the entries in the STLF buffer (or store queue 400, or LS2 buffer 62.) Instead, the Examiner's citations describe that a comparison is made between an index portion of a load address and an index portion of an entry in a store queue or an LS2 buffer to determine whether to forward data from the entry. Applicants assert that performing a comparison between the index portions of the load address and an entry corresponding to a store operation is clearly not the same as indexing into the indexed STLF buffer using the generated index to select an entry in an STLF buffer, as would be understood by one skilled in the art (i.e., using the generated index as an index into the indexed STLF buffer).

Also, as noted above, there is nothing in this citation or elsewhere in Hughes that describes generating an index dependent on at least a portion of an address of a load operation, as recited in claim 1. Instead, Hughes only teaches using an index portion of an address to compare a load operation to one or more entries in store queue 400 or LS2 62. This is described on p. 7, lines 4-9:

> If the load index and a store index of a store represented in store queue 400 match, the load and that store are a hit, and the way indications of the load and that store match, hit control circuit 402 causes data to be forwarded from store queue 400 for the load. More particularly, hit control circuit 402 may signal store queue 400 with an indication of the entry number of the entry being hit, and store queue 400 may provide the data from that entry for forwarding in place of the cache data from the data cache.

This citation describes a system and method clearly different from Applicants' claimed invention by describing that the address index portions (among other elements) for a load operation and one or more store operations are used to perform a comparison, and then an indication of an entry number in store queue 400 is provided if one of the entries of store queue 400 matches the load operation. There is nothing in this citation or elsewhere in Hughes that teaches or suggests that this "indication of the entry number of the entry" corresponds to an index portion of the load or store address (as suggested by the Examiner) or to an index generated by the load/store unit dependent on at least a portion of an address of the load operation, as in Applicants' claim 1.

In the Advisory Action mailed August 21, 2006 the Examiner notes that Applicants state that a comparison between the index portions of the load address and an entry corresponding to a store operation is clearly not the same as using an index to select an entry. The Examiner submits that the index is used, depending on whether or not it matches, to determine whether or not to select that particular entry for forwarding and the entry in question is, in fact, one of a plurality of entries; therefore, the limitations of the claim (specifically, "to use the index to select one of the plurality of entries") are met. Applicants have amended claims 1, 14, and 20 to further clarify that the generated index of Applicants' invention is not merely "used to select" an entry, but that it is used to index into an indexed STLF buffer to select an entry. Applicants assert that as amended,

the independent claims clearly distinguish over the cited art, since the cited reference uses an index portion of an address in a comparison operation to select an entry in a buffer rather than using it to index into an indexed STLF buffer.

Applicants also note that the Examiner has not included any specific remarks regarding the limitation "*the load store unit is configured to generate an index dependent on at least a portion of an address of a load operation*" in the Advisory Action, the nor do the cited references teach or suggest this limitation. As discussed above, Hughes describes using an index portion of an address, but not a load store unit configured to generate an index, as in Applicants' claim 1.

Therefore, for at least the reasons above, the rejection of claim 1 is not supported by the cited art and Applicants respectfully request the withdrawal thereof.

Independent claims 14 and 20 each include a limitation involving generating an index and each has been similarly amended to clarify the use of the generated index to index into an indexed STLF buffer to select an entry. Therefore, the arguments presented above regarding claim 1 apply with equal force to these claims as well.

Regarding claim 5, contrary to the Examiner's assertion, Hughes fails to teach or suggest *the load store unit is configured to select which one of the plurality of entries to allocate to a store operation by generating an additional index dependent on at least a portion of an address of the store operation.* The Examiner cites FIG. 1 (ADDR – Tag) as teaching this limitation. However, the address tag portion of an entry in store queue 400 is not described as an additional index used to select which one of the plurality of entries to allocate to a store operation, nor is it generated by the load store unit. Instead, as stated on page 6, lines 18-19, "The address tag portion is the portion of the address which is stored as a tag by the data cache..." Thus, this tag is clearly not generated by the load store unit and used in selecting an entry in a STLF buffer, as required by Applicants' claim 5. Therefore, Hughes cannot be said to anticipate claim 5.

For at least the reasons above, the rejection of claim 5 is unsupported by the cited art and removal thereof is respectfully requested. Claim 16 and 24 include limitations similar to claim 5, and so the arguments presented above apply with equal force to these claims, as well.

Regarding claim 6, contrary to the Examiner's assertion, Hughes fails to teach or suggest *the load store unit is configured to generate the additional index dependent on both the at least a portion of the address of the store operation and a number of bytes of data operated on by the store operation, and wherein the load store unit is configured to generate the index dependent on both the at least a portion of the address of the load operation and a number of bytes of data operated on by the load operation.* As discussed above regarding claims 1 and 5, Hughes fails to teach or suggest generating an index or generating an additional index, at all. The Examiner cites page 8, lines 10-13, as teaching these limitations, "The Examiner asserts that the data size is used as an index to determine if a TRAP signal is to be generated due to a load requesting a larger block of data than was previously stored at that memory location." However, this citation states, "Hit control circuit 402 may use the offset and size information to determine whether or not to cause the forwarding of data stored in store queue 400 for the load (in addition to the index comparisons, hit bits, and way indications described above)." The offset and size information are used in additional comparison operations to determine if there is a match between a load operation and a previous store operation, not to generate an index for selecting an entry in a STLF buffer. This citation has nothing to do with generating an index, much less with the specific details for generating two different indexes that are recited in claim 6. Furthermore, the Examiner's remarks regarding a TRAP signal have absolutely no basis in the cited art, as Hughes does not include a description of a TRAP signal at all. These remarks appear to be copied from the Examiner's previous rejection of claim 6 as being anticipated by Webb and are therefore improper to include in the rejection of claim 6 as being anticipated by Hughes. In addition, the use of size data to determine if a TRAP signal is generated has nothing to do with the limitations of claim 6, which are directed toward generating indexes used for selecting an entry in a STLF buffer. Therefore Hughes clearly does not anticipate claim 6.

For at least the reasons above, the rejection of claim 6 is unsupported by the cited art and removal thereof is respectfully requested. Claim 17 and 25 include limitations similar to claim 6, and so the arguments presented above apply with equal force to these claims, as well.

Regarding claim 8, contrary to the Examiner's assertion, Hughes fails to teach or suggest the additional index comprises a portion of the address targeted by the store operation. The Examiner again cites FIG. 1 (ADDR – tag) as teaching this limitation. However, as discussed above, Hughes does not teach generating an additional index at all, much less one comprising a portion of the address targeted by the store operation. The address tag in FIG. 1 is the portion of the store address that is stored as a tag by the data cache, not an additional index used in selecting an entry in a STLF buffer. In addition, this tag is clearly not generated by the load store unit, as required by Applicants' claims 5 and 8.

For at least the reasons above, the rejection of claim 8 is unsupported by the cited art and removal thereof is respectfully requested. Claim 27 includes limitations similar to claim 8, and so the arguments presented above apply with equal force to this claim, as well.

Regarding claim 10, contrary to the Examiner's assertion, Hughes fails to teach or suggest *the STLF checker is configured to perform an associative address comparison to identify all issued store operations targeting a same address as the load operation and to implement a find-first algorithm to select a youngest issued store operation that is older than the load operation.* The Examiner cites page 8, lines 14-17 as teaching these limitations. However, while this passage states, "Hit control circuit 402 may determine the youngest (most recently executed) store in program order among the stores corresponding to entries which are hit and may forward the data from that entry," there is nothing in Hughes that discloses the specific limitations of claim 10. For example, there

is no mention of implementing a find-first algorithm, as recited in claim 10. Therefore Hughes cannot be said to anticipate claim 10.

For at least the reasons above, the rejection of claim 10 is unsupported by the cited art and removal thereof is respectfully requested. Claim 28 includes limitations similar to claim 10, and so the arguments presented above apply with equal force to this claim, as well.

Regarding claim 11, contrary to the Examiner's assertions, Hughes fails to teach or suggest *the STLF checker is configured to replay the load operation if the STLF checker identifies incorrect operation of the STLF buffer.* The Examiner cites page 5, lines 15-18 as teaching this limitation. However, this passage does not describe replaying an operation (e.g., by providing a signal to a scheduler or by other means suitable to cause the instruction to be replayed). In fact, there is no mention of an instruction or operation being replayed anywhere in Hughes. Instead, this passage states (in part), "...the data cache is not forwarding data from the cache for the load if the load is miss, and thus store data from store queue 400 need not be forwarded. The load may be reattempted after the data cache is filled with the cache line read by the load (or during the writing of fill data into the cache), and any stores to that cache line may become hits during the cache fill." This passage describes what happens in response to a cache miss for the load operation, not in response to an STLF checker identifying incorrect operation of the STLF buffer. Furthermore, the response does not include replaying the load operation, but may include filling the data cache with a cache line read by the load or with fill data, which may result in a cache hit if the load is reattempted. Therefore, Hughes clearly does not anticipate claim 11.

For at least the reasons above, the rejection of claim 11 is unsupported by the cited art and removal thereof is respectfully requested. Claim 19 and 29 include limitations similar to claim 11, and so the arguments presented above apply with equal force to these claims, as well.

Regarding claim 12, contrary to the Examiner's assertion, Hughes fails to teach or suggest *the STLF checker is configured to replay one or more additional operations that are dependent on the load operation if the STLF checker detects incorrect operation of the STLF buffer.* The Examiner again cites page 5, lines 15-18 as teaching this limitation. However, as discussed above, this passage does not describe replaying a load operation in response to the STLF checker detecting incorrect operation of the STLF buffer, much less replaying one or more additional operations, as the Examiner suggests. Instead, it describes actions that may occur following a miss in the data cache for a load instruction. The Examiner asserts, "...after an initial load instruction is attempted and fails, any other instructions which have been issued which depend on the load instruction for data must be replayed once the data has been made available through replay of the load operation." This is purely speculation by the Examiner. Applicants assert that there are many different ways to recover from a data cache miss for a load instruction, not all of which necessarily involve replaying the load instruction or any dependent instructions. Furthermore, a data cache miss for a load operation is not necessarily caused by incorrect operation of a STLF buffer. The Examiner has not cited anything in Hughes to teach or suggest that a load operation and one or more additional operations are replayed in response to incorrect operation of a STLF buffer (or Hughes' store queue 400 or LS2 62), nor is there anything in Hughes describing such replays. Therefore, Hughes clearly does not anticipate claim 12.

For at least the reasons above, the rejection of claim 12 is unsupported by the cited art and removal thereof is respectfully requested. Claim 30 includes limitations similar to claim 12, and so the arguments presented above apply with equal force to this claim, as well.

Regarding claim 13, contrary to the Examiner's assertion, Hughes fails to teach or suggest *the load store unit is configured to identify the result of the load operation as a speculative value in response to forwarding the data in the one of the plurality of entries included in the STLF buffer as the result of the load operation; wherein if the STLF checker verifies that the STLF buffer operated correctly for the load operation, the load*

store unit is configured to indicate that the result of the load operation is not speculative. The Examiner cites the abstract and claims 1-4 of Hughes as teaching these limitations. FIG. 1 illustrates store queue 400. There is nothing in FIG. 1, or its description, that teaches or suggests these limitations of claim 13. In addition, claims 1-4 recite limitations regarding asserting a forward signal and forwarding store data from an entry of store queue 400 if there is a match between an index portion of a load address and an index portion of the entry, and if a hit indication of the entry indicates a hit in a data cache. None of the limitations of claims 1-4 has anything to do with identifying a result of a load operation as speculative or with indicating that the result of the load operation is not speculative if the STLF checker verifies correct operation of the STLF buffer for the load operation. Therefore Hughes clearly does not anticipate claim 13.

For at least the reasons above, the rejection of claim 13 is unsupported by the cited art and removal thereof is respectfully requested. Claim 31 includes limitations similar to claim 13, and so the arguments presented above apply with equal force to this claim, as well.

The Examiner rejected claims 1-3, 5-6, 8-22, 24-25 and 27-31 under 35 U.S.C. § 102(b) as being anticipated by Webb et al. (U.S. Patent 6,360,314) (hereinafter "Webb"). Applicants traverse this rejection for at least the following reasons.

Regarding claim 1, contrary to the Examiner's assertion, Webb fails to teach or suggest *a load store unit configured to store information associated with load and store operations dispatched by the dispatch unit, wherein the load store unit includes an indexed STLF (Store-to-Load Forwarding) buffer, wherein the indexed STLF buffer includes a plurality of entries; wherein the load store unit is configured to generate an index dependent on at least a portion of an address of a load operation, to index into the indexed STLF buffer using the generated index to select one of the plurality of entries.* First, Webb does not disclose a load store unit configured to store information associate with load and store operations, nor including an indexed Store-to-Load Forwarding (STLF) buffer. The Examiner refers to FIG. 4, including buffer 428 and queue 426 as

teaching these limitations. However, FIG. 4 clearly illustrates that store data buffer 428 and store queue 426 are components of data cache subsystem 420, and not load/store unit 418.

Further regarding claim 1, Webb does not disclose the load store unit generating an index and indexing into the indexed STLF buffer using the generated index to select one of the entries in the STLF buffer. The Examiner cites column 5, lines 5-8 and 19-22, as teaching these limitations. However, there is nothing in these citations that teaches that the load store unit, or any other apparatus, is configured to **generate** an index. They only describe the use of an index. Furthermore, the index referred to in these citations is not a generated index used to index into an indexed STLF buffer, or to Webb's store queue 426 and store data buffer 428, which the Examiner equates with Applicants' STLF buffer. Instead, these citations describe indexing into dcache unit 430, and its components (tag store 432 and data store 434.) Therefore, Webb clearly does not teach these limitations of Applicants' claim 1.

As the Examiner is no doubt aware, anticipation requires the presence in a single prior art reference disclosure of each and every limitation of the claimed invention, arranged as in the claim. M.P.E.P 2131; *Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 221 USPQ 481, 485 (Fed. Cir. 1984). The **identical** invention must be shown in as complete detail as is contained in the claims. *Richardson v. Suzuki Motor Co.*, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). As discussed above, Webb fails to disclose *a load store unit configured to store information associated with load and store operations dispatched by the dispatch unit, wherein the load store unit includes an indexed STLF (Store-to-Load Forwarding) buffer, wherein STLF buffer includes a plurality of entries; wherein the load store unit is configured to generate an index dependent on at least a portion of an address of a load operation, to index into the indexed STLF buffer using the generated index to select one of the plurality of entries.* Therefore, Webb cannot be said to anticipate claim 1.

In the Response to Arguments section of the Final Action, the Examiner states that Applicants' arguments were not found persuasive because Webb's invention contains a "Store-to-Load Forwarding buffer" and that this buffer and its associated logic constitute part of the load/store unit as they all assist in performing load and store operations. For the reasons stated above, Applicants disagree.

For at least the reasons above, the rejection of claim 1 is not supported by the cited art and removal thereof is respectfully requested. Independent claim 14 includes limitations similar to claim 1, and so the arguments presented above apply with equal force to this claim, as well.

Regarding independent claim 20, contrary to the Examiner's assertion, Webb fails to teach or suggest *generating an index corresponding to the address* and *indexing into an indexed STLF (Store-to-Load Forwarding) buffer using the generated index to select an entry from a plurality of entries included in an indexed STLF buffer.* The Examiner cites column 5, lines 5-22 as teaching this limitation. However, as discussed above regarding claim 1, Webb discloses neither generating an index, nor indexing into an indexed STLF buffer using the generated index to select an entry in a STLF buffer, as recited in claim 20.

For at least the reasons above, the rejection of claim 20 is not supported by the cited art and removal thereof is respectfully requested.

Regarding claim 5, contrary to the Examiner's assertion, Webb fails to teach or suggest *the load store unit is configured to select which one of the plurality of entries to allocate to a store operation by generating an additional index dependent on at least a portion of an address of the store operation.* The Examiner again cites column 5, lines 5-8 and 19-23 as teaching this limitation. However, as discussed above, there is nothing in Webb that teaches the load store unit is configured to generate an index at all, much less two indices, and these citations describe indexing into dcache 430, not into a STLF buffer, or store queue 426 and store data buffer 428.

For at least the reasons above, the rejection of claim 5 is not supported by the cited art and removal thereof is respectfully requested. Claims 16 and 24 include limitations similar to claim 5, and so the arguments presented above apply with equal force to these claims, as well. Applicants note that the Examiner rejected claim 22 along with claim 5 in his remarks. Applicants assume that this was a typographical error, as claim 24 (and not claim 22) recites limitations similar to those in claim 5.

Regarding claim 6, contrary to the Examiner's assertion, Webb fails to teach or suggest *the load store unit is configured to generate the additional index dependent on both the at least a portion of the address of the store operation and a number of bytes of data operated on by the store operation, and wherein the load store unit is configured to generate the index dependent on both the at least a portion of the address of the load operation and a number of bytes of data operated on by the load operation.* The Examiner again cites column 5, lines 5-8 and 19-23, as teaching that the load store unit is configured to generate this index. However, as discussed above regarding claim 5, Webb does not teach this limitation. The Examiner cites column 6, lines 51-59, as teaching that the index is generated dependent on the number of bytes of data operated on by the store operation. While this citation describes a comparison between the size field 48 of a store queue entry and the size information of an issuing load, it does not describe generating an index into a STLF buffer dependent on this information. The Examiner asserts that the "data size is used as an index to determine if a TRAP signal is to be generated...". However, as would be understood by one of ordinary skill in the art, using a comparison of two data size values to generate a signal is clearly not the same as generating an index dependent on a data size value.

In the Response to Arguments section of the Final Action, the Examiner submits that these arguments were not found persuasive. The Examiner states, "As previously cited and applied, the examiner clarifies that the size of the pending load *is* used as an index to verify the proper data is being loaded from the buffer" and that, "the size of the load is inherently passed into the buffer logic from the load/store unit and compared, as

described in column 6, lines 51-59." Applicants note that claim 6 has nothing to do with "verifying the proper data is being loaded from the buffer" but instead is directed toward indexes used to select an entry in a STLF buffer. The Examiner further states, "both the address portion and the data access size combined constitute the index described in claim 6." Applicants disagree. Nothing in Hughes teaches or suggests these elements of a load operation, taken separately or in combination with each other, is an index generated by the load store unit and used to select an entry in a STLF buffer.

The Examiner notes, "Webb performs a comparison of the size as an index in the same manner as the applicant describes in paragraph 36 on page 12 of the specification" (emphasis added). Applicants note that Webb does not describe performing a comparison to generate an index for selecting an entry in a STLF buffer, as the Examiner suggests, or using the size itself to select an entry in a STLF buffer. Applicants also note that paragraph 36 of Applicants' specification describes a comparison between the address or size of a selected entry (i.e., one selected according to a first generated index) and the address or size of a load operation. It does not describe that this comparison constitutes generating an index or selecting an entry, as the Examiner suggests. Paragraph 37, however, goes on to describe an example in which size data may be used in generating an index. This passage describes that an index may be "generated by performing some transformation function on the address of an operation. Indexes may be generated such that any given address and data size pair maps to a single index..." There is nothing in Hughes that describes generating an index dependent on the number of bytes of data operated on by a load or store operation, as required by claim 6, in this or any other manner.

For at least the reasons above, the rejection of claim 6 is not supported by the cited art and removal thereof is respectfully requested. Claims 17 and 25 include limitations similar to claim 6, and so the arguments presented above apply with equal force to these claims, as well.

Regarding claim 8, contrary to the Examiner's assertion, Webb fails to teach or suggest *the additional index comprises a portion of the address targeted by the store operation*. The Examiner again cites column 5, lines 19-22, as teaching this limitation. However, this citation describes indexing (using a portion of an address) into dcache unit 430, not into a STLF buffer, or into Webb's store queue 426 and store data buffer 428. Using an address to index into dcache unit 430 teaches nothing about the composition of the additional index (for selecting an entry in the STLF buffer) referred to in Applicants' claim 8.

In the Response to Arguments section, the Examiner "points to FIG. 4 indicating that the additional index generated by the load/store unit is connected to the buffer store queue 426 in addition to dcache 430, as indicated by lines 442 and 446." However, FIG. 4 illustrates that the same index as the one the Examiner refers to in his rejection of claim 1 (described in column 5, lines 5-8 and 19-22) is connected to both the dcache and the buffer store queue. Therefore, this cannot be considered "an additional index" as the Examiner suggests.

For at least the reasons above, the rejection of claim 8 is not supported by the cited art and removal thereof is respectfully requested. Claim 27 includes limitations similar to claim 8, and so the arguments presented above apply with equal force to this claim, as well.

Regarding claim 9, contrary to the Examiner's assertion, Webb fails to teach or suggest *the load store unit further comprises a STLF checker configured to verify operation of the STLF buffer*. The Examiner cites column 7, lines 5-8, as teaching this limitation, and asserts that Webb's invention contains a unit which verifies operation, specified as the apparatus disclosed in column 7, lines 5-8. However, the Examiner's citation describes only that, "The present invention provides a methodology and apparatus for determining which of the multiple stores should be used in bypassing the dcache unit 430." This does not describe a unit *configured to verify operation of the STLF buffer*, as recited in claim 9, but describes only one of the operations of "a

methodology and apparatus" for choosing which of multiple stores should be used. Applicants assert that choosing one of multiple stores has nothing to do with verifying operation of the STLF buffer, as recited in claim 9.

In the Response to Arguments section of the Final Action, the Examiner submits, "there must inherently exist logic to verify that the correct entry is forwarded when multiple stores are pending for the same address location. If the youngest store is not forwarded upon a subsequent load, incorrect operation may result." Applicants note that the data bypass method and apparatus described at the Examiner's citation in Webb includes logic to "attempt to provide the data of the most recent store to a subsequently issuing load and thereby avoid getting older data to the load," (column 7, lines 18-20). Attempting to provide the correct data is clearly not equivalent to the limitation of Applicants' claim 9, which recites *the load store unit further comprises a STLF checker configured to verify operation of the STLF buffer*. Furthermore, Applicants remind the Examiner that "To establish inherency, the extrinsic evidence 'must make clear that the missing descriptive matter is **necessarily present** in the things described in the reference, and that it would be so recognized by persons of ordinary skill. Inherency, however, may not be established by probabilities or possibilities. The mere fact that a certain thing may result from a given set of circumstances is not sufficient.'" *In re Robertson*, 169 F.3d 743, 745, 49 USPA2d 1949, 1950-51 (Fed. Cir. 1999) (emphasis added). The Examiner has not cited anything in Webb that teaches or suggests *the load store unit further comprises a STLF checker configured to verify operation of the STLF buffer*.

For at least the reasons above, the rejection of claim 9 is not supported by the cited art and removal thereof is respectfully requested. Claim 18 includes limitations similar to claim 9, and so the arguments presented above apply with equal force to this claim, as well.

Regarding claim 11, contrary to the Examiner's assertion, Webb fails to teach or suggest *the STLF checker is configured to replay the load operation if the STLF checker identifies incorrect operation of the STLF buffer*. The Examiner cites column 1, lines 34-

38 as teaching this limitation and asserts that if a load operation does not result in data being provided from the cache, the memory load is replayed to the main memory. This is incorrect. First, this citation teaches nothing about replaying a load operation. It merely describes that the data for a load is retrieved from main memory, instead of from the cache, if it is not found in the cache. Furthermore, this citation has nothing to do with a STLF checker identifying incorrect operation of the STLF buffer. In fact, this citation has nothing to do with the STLF buffer at all. Instead, as stated above, this citation describes only that if data is not found in the cache, it is retrieved from main memory.

In the Response to Arguments section of the Final Action, the Examiner submits, "when a load misses after a first attempt in the buffer of Webb's system, the load must be replayed (attempted again) to another memory device (in this case, the dcache unit 430). There is nothing is applicant's specification defining the term "replay" and hence, it has been awarded its broadest reasonable definition." Applicants disagree with the Examiner's interpretation of the term "replay" and also with his assertion that Applicants' specification does not include anything defining this term. Applicants note that the specification includes a specific example of how a replay may be caused by the STLF checker on page 16, lines 17-18, "The STLF checker 303 may cancel load operations that either incorrectly forwarded in STLF buffer 305 or which incorrectly did not forward in STLF buffer 305. In such situations, the STLF checker 303 may cause the load operation to be replayed (e.g., by providing a signal to the scheduler 118)." Thus, it is clear from Applicants' specification that the term "replay" was intended to refer to situations in which a load operation is canceled and then rescheduled for execution (i.e., the current instance of the load operation is cancelled and another instance of the load operation is scheduled for execution) rather than to situations in which a current load operation may search more than one level of memory in an attempt to fulfill the load.

For at least the following reasons, the rejection of claim 11 is not supported by the cited art and removal thereof is respectfully requested. Claims 19 and 29 include limitations similar to claim 11, and so the arguments presented above apply with equal force to these claims, as well.

Regarding claim 12, contrary to the Examiner's assertion, Webb fails to teach or suggest *the STLF checker is configured to replay one or more additional operations that are dependent on the load operation if the STLF checker detects incorrect operation of the STLF buffer.* The Examiner cites column 7, lines 63-65, as teaching this limitation and asserts that after the in-flight instructions are killed they must inherently be reissued. The Examiner contends that if the instructions are not reissued, the program may produce undesired output or simply cease operation. The Examiner's citation describes that a TRAP signal 454 is provided by store queue 426 to indicate that the bypass mechanism did not provide data when it should have and therefore, the in-flight instructions should be killed. Applicants assert that the system of Webb could respond to the TRAP signal in any number of ways to produce the correct output or prevent the program from ceasing operation. There is nothing in Webb that teaches or suggests that the response to this signal is to *replay one or more additional operations that are dependent on the load operation*, as recited in claim 12. As there is no detailed description of the scheduling of instructions in Webb, or of any **replay** mechanism at all, the Examiner's assertion that one or more additional operations that are dependent on the load operation must be replayed is mere speculation, not a necessary condition of operation.

In the Response to Arguments section of the Final Action, the Examiner submits, "inflight instructions which are killed must inherently be re-issued if proper program outcome is to be achieved. As described by Webb, when the TRAP signal is issued in the instance described in col. 7, it signals that the load operation did not provide data when it should have (column 7, line 34) resulting in incorrect and/or unavailable data. A program contains instructions in a certain order designed to achieve a pre-defined function. If a system were to kill the inflight instructions following a data miss (like that described in col. 7) without reissuing them, the function of the program may not be achieved."

Applicants again remind the Examiner that, "To establish inherency, the extrinsic evidence 'must make clear that the missing descriptive matter is **necessarily present** in the things described in the reference, and that it would be so recognized by persons of

ordinary skill. Inherency, however, may not be established by probabilities or possibilities. The mere fact that a certain thing may result from a given set of circumstances is not sufficient.'" *In re Robertson*, 169 F.3d 743, 745, 49 USPA2d 1949, 1950-51 (Fed. Cir. 1999) (emphasis added). As discussed above, the system of Webb would not **necessarily** need to *replay one or more additional operations that are dependent on the load operation if the STLF checker detects incorrect operation of the STLF buffer* in order to recover from instructions being killed and/or to achieve the function of the program.

For at least the reasons above, the rejection of claim 12 is not supported by the cited art and removal thereof is respectfully requested. Claim 30 includes limitations similar to claim 12, and so the arguments presented above apply with equal force to this claim, as well.

Regarding claim 13, contrary to the Examiner's assertion, Webb fails to teach or suggest *the load store unit is configured to identify the result of the load operation as a speculative value in response to forwarding the data in the one of the plurality of entries included in the STLF buffer as the result of the load operation; wherein if the STLF checker verifies that the STLF buffer operated correctly for the load operation, the load store unit is configured to indicate that the result of the load operation is not speculative.* The Examiner asserts that while a load operation is pending in the store buffer/queue (while the address lookup is operating) the operation is speculative, as no data has yet been forwarded. The Examiner further asserts that when the index has been calculated, and the apparatus determines the entry to be the youngest and of the correct size, the data is forwarded and is then no longer considered speculative. The Examiner is incorrect. First, claim 13 clearly recites that the load store unit identifies the result of the load operation as speculative in response to forwarding the data in the one of the plurality of entries included in the STLF buffer as the result of the load operation, that is, the result is identified as speculative **after the load operation and after the data has been forwarded**, not while the load operation is pending, as the Examiner asserts. The Examiner is not describing a form of speculation, as recited in claim 13 and as would be understood by one of ordinary skill in the art, nor does Webb disclose this speculation.

Instead, rather than describing a case in which the results of the load operation are speculative, the Examiner appears to be describing that the load operation itself is speculative, which has nothing to do with Applicants' claims.

Further regarding claim 13, Webb does not disclose that the load store unit, or any other apparatus, is configured to identify the result of the load operation as a speculative value. Even if the Examiner's assertions regarding speculation were correct, which they are not, Webb does not disclose providing any indication that the result of the load operation is speculative or not speculative, as recited in claim 13.

For at least the reasons above, the rejection of claim 13 is not supported by the cited art and removal thereof is respectfully requested. Claim 31 includes limitations similar to claim 13, and so the arguments presented above apply with equal force to this claim, as well.

Applicants note that the Examiner did not repeat his previous rejection of claims 13 and 31 in the Final Action and did not include any remarks regarding these claims in the Response to Arguments section of the Final Action. Therefore, Applicants assume that the Examiner considers the arguments above regarding the rejection of claims 13 and 30 to be persuasive and request removal of the rejection of claims 13 and 30 under 35 U.S.C. § 102(b) as being anticipated by Webb.

Section 103(a) Rejections:

The Examiner rejected claims 4 and 23 under 35 U.S.C. § 103(a) as being unpatentable over Hughes, and claims 7 and 26 as being unpatentable over Hughes in view of Hennessy (Computer Organization and Design). Applicants traverse these rejections for at least the following reasons.

Regarding claim 4, contrary to the Examiner's assertion, Hughes fails to teach or suggest *each of the plurality of entries in the STLF buffer has a capacity to store a maximum amount of data that can be written by a store operation*. The Examiner admits that Hughes fails to disclose this limitation, "Hughes does not disclose the data bus width of the processor of his invention." The Examiner takes Official Notice that data buses of size 8, 16, 32, or 64 bits are extremely well known in the art and that with any of these data buses in place in Hughes invention, the data buffer would be able to hold at least the maximum amount of data specified by a data store operation. Applicants assert however, that the maximum amount of data that can be written by a store operation is not necessarily the same as, or dependent on, the data bus width, but instead may be dependent on the instruction set architecture (i.e., the instruction sets of different processors may provide for store operations involving different numbers of bytes of data and/or the implementation of the instruction set may provide for different numbers of bus cycles for each store operation). Therefore, Applicants assert that the Examiner's Official Notice does not teach or suggest the limitations of claim 4.

For at least the reasons above, the rejection of claim 4 is not supported by the cited art and removal thereof is respectfully requested. Claim 23 includes limitations similar to claim 4, and so the arguments presented above apply with equal force to this claim, as well.

Regarding claim 7, contrary to the Examiner's assertion, Hughes in view of Hennessy fails to teach or suggest *the additional index is generated by right-shifting a lower portion of the address targeted by the store operation by an amount equal to a logarithm in base two of the number of bytes of data operated on by the store operation*. The Examiner admits that Hughes fails to disclose this limitation and relies on Hennessey to teach it, "Hennessey discloses indexing a cache by means of the lower portion of an address, minus the appropriate offset for minimum memory access size (byte, in Hennessey's case)... Right shifting is an extremely well-known method of eliminating undesired bits to the right of desired bits. Hennessey teaches that removing the two bits for the byte offset reduces the total cache size, as fewer bits must be kept in the tag entry

field of each entry. A smaller cache size takes up less space on chip and is less expensive to implement." The Examiner submits that it would have been obvious to one of ordinary skill in the art at the time of invention to have included the method of generating a cache index disclosed by Hennessey in Hughes' invention for the benefit of reducing the necessary cache size. Applicants assert, however, that claim 7 has nothing to do with indexing a cache or reducing a cache size, but is directed toward a specific method for generating an additional index for selecting an entry in the STLF buffer. As discussed above, Hughes does not teach generating this additional index, so one would not have any reason to explore ways to generate such an index. Furthermore, if the method of Hennessey were applied to Hughes' invention, the result would not teach the limitations of claim 7. Instead, they would result in a change in the way the data cache, not the store queue, was organized and indexed.

For at least the reasons above, the rejection of claim 7 is not supported by the cited art and removal thereof is respectfully requested. Claim 26 includes limitations similar to claim 7, and so the arguments presented above apply with equal force to this claim, as well.

The Examiner rejected claims 4 and 23 under 35 U.S.C. § 103(a) as being unpatentable over Webb, and claims 7 and 26 as being unpatentable over Webb in view of Hennessy ("Computer Organization and Design"). Applicants traverse this rejection for at least the following reasons.

Regarding claim 4, contrary to the Examiner's assertion, Webb fails to teach or suggest *each of the plurality of entries in the STLF buffer has a capacity to store a maximum amount of data that can be written by a store operation.* The Examiner admits that Webb fails to disclose this limitation and notes that while Webb discloses the data entry to hold any of a quadword, longword, word, or byte (column 4, lines 66-67), it does not disclose the data bus width of the processor of his invention. The Examiner takes Official Notice that data buses of size 8, 16, 32, or 64 bits are extremely well known in the art and that with any of these data buses in place in Webb's invention, the data buffer

would be able to hold at least the maximum amount of data specified by a data store operation. Applicants note that claim 4 does not recite a limit on the data bus width of the invention, but instead recites *each of the plurality of entries in the STLF buffer has a capacity to store a maximum amount of data that can be written by a store operation.* Applicants assert that the maximum amount of data that can be written by a store operation is not necessarily the same as, or dependent on, the data bus width, but instead may be dependent on the instruction set architecture (i.e., the instruction sets of different processors may provide for store operations involving different numbers of bytes of data and/or the implementation of the instruction set may provide for different numbers of bus cycles for each store operation). Therefore, Applicants assert that the Examiner's Official Notice does not teach or suggest the limitations of claim 4.

The Examiner submits, "A data bus having a given size less than or equal to 64 bits is beneficial in a processor in that costs of implementation are limited. Larger data buses require processor components to also grow in size, increasing processor area, power consumption and cost. Implementing a data bus less than or equal to 64 bits in Webb's invention would have been obvious at the time of invention to one of ordinary skill in the art for the benefit of limiting costs, size and power consumption." As noted above, however, claim 4 does not recite "a data bus less than or equal to 64 bits" or anything about a data bus width at all. Even if the data bus of Webb were less than or equal to 64 bits, this would teach nothing about the capacity of the entries of an STLF buffer, as recited in claim 4.

For at least the reasons above, the rejection of claim 4 is not supported by the cited art and removal thereof is respectfully requested. Claim 23 includes limitations similar to claim 4, and so the arguments presented above apply with equal force to this claim, as well.

Regarding claim 7, contrary to the Examiner's assertion, Webb in view of Hennessy fails to teach or suggest *the additional index is generated by right-shifting a lower portion of the address targeted by the store operation by an amount equal to a*

logarithm in base two of the number of bytes of data operated on by the store operation. The Examiner admits that Webb fails to disclose this limitation and relies on Hennessey to teach it, "Hennessey discloses indexing a cache by means of the lower portion of an address, minus the appropriate offset for minimum memory access size (byte, in Hennessey's case)... Right shifting is an extremely well-known method of eliminating undesired bits to the right of desired bits. Hennessey teaches that removing the two bits for the byte offset reduces the total cache size, as fewer bits must be kept in the tag entry field of each entry. A smaller cache size takes up less space on chip and is less expensive to implement." The Examiner submits that it would have been obvious to one of ordinary skill in the art at the time of invention to have included the method of generating a cache index disclosed by Hennessey in Webb's invention for the benefit of reducing the necessary cache size. Applicants assert, however, that claim 7 has nothing to do with indexing a cache or reducing a cache size, but is directed toward a specific method for generating an additional index for selecting an entry in the STLF buffer. As discussed above, Webb does not teach generating this additional index, so one would not have any reason to explore ways to generate such an index. Furthermore, if the method of Hennessey were applied to Webb's invention, the result would not teach the limitations of claim 7. Instead, they would result in a change in the way the data cache, not the store queue, was organized and indexed.

For at least the reasons above, the rejection of claim 7 is not supported by the cited art and removal thereof is respectfully requested. Claim 26 includes limitations similar to claim 7, and so the arguments presented above apply with equal force to this claim, as well.

Applicants assert that numerous other ones of the dependent claims recite further distinctions over the cited art. Applicants traverse the rejection of these claims for at least the reasons given above in regard to the claims from which they depend. However, since the rejections have been shown to be unsupported for the independent claims, a further discussion of the dependent claims is not necessary at this time. Applicants reserve the right to present additional arguments.

CONCLUSION

Applicants submit the application is in condition for allowance, and prompt notice to that effect is respectfully requested.

If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5500-89400/RCK.

Also enclosed herewith are the following items:

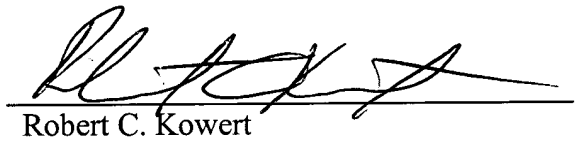
☒ Return Receipt Postcard

☐ Petition for Extension of Time

☐ Notice of Change of Address

☒ Request for Continued Examination

Respectfully submitted,



Robert C. Kowert
Reg. No. 39,255
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8850

Date: September 5, 2006